FIG. 1
Prior Art

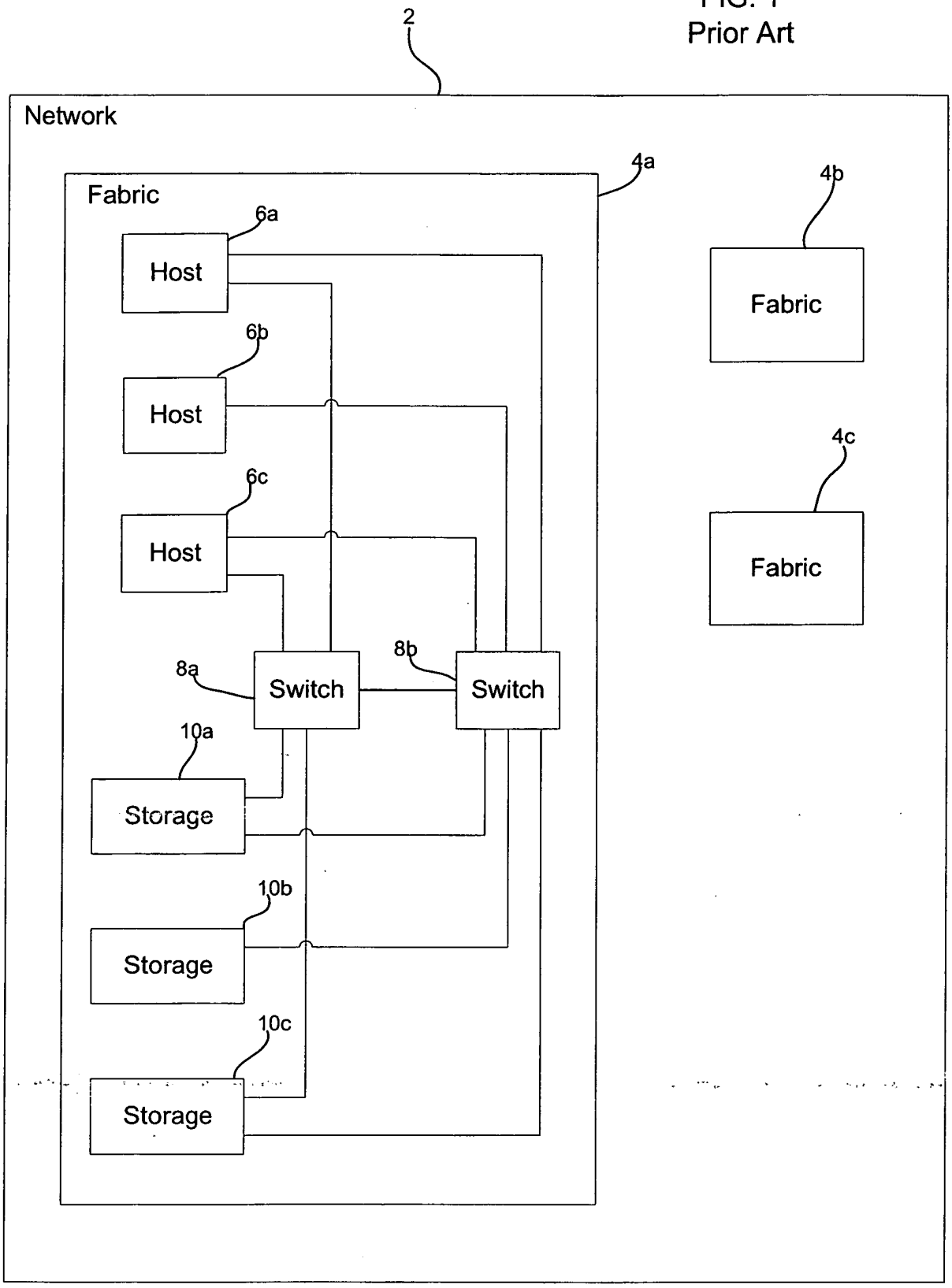


FIG. 2

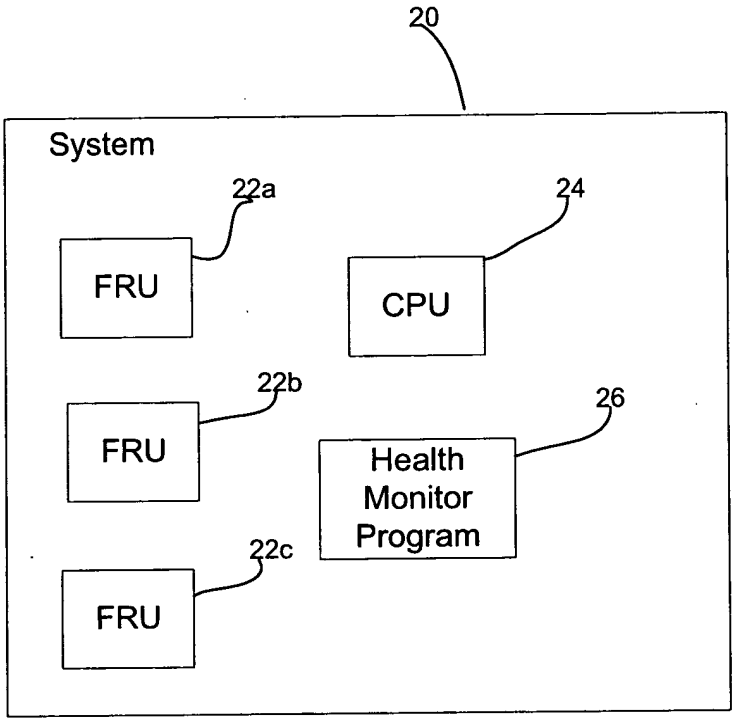

20
System
22a
24
FRU
CPU
22b
26
FRU
Health
Monitor
Program
22c
FRU

FIG. 3

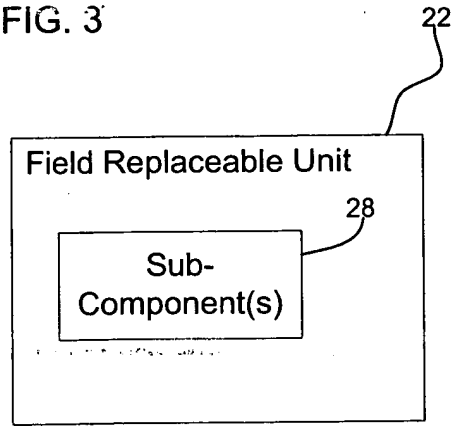

22
Field Replaceable Unit
28
Sub-
Component(s)

FIG. 4

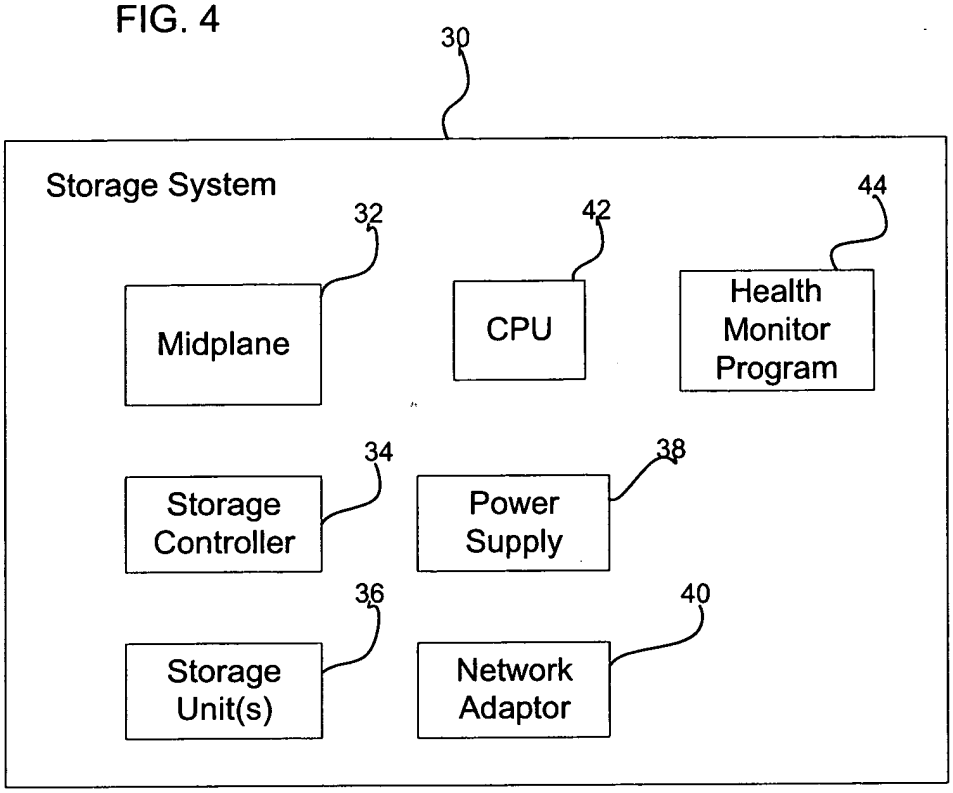

30
Storage System
32
Midplane
42
CPU
44
Health Monitor Program
34
Storage Controller
38
Power Supply
36
Storage Unit(s)
40
Network Adaptor

FIG. 5

Truth Table for Power Supply

PS_State	*Temp_Status*	*Battery_Status*	*Fan_Status*	*Output State*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	D
0	1	0	0	D
0	1	0	1	D
0	1	1	0	D
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	D
1	1	0	0	D
1	1	0	1	D
1	1	1	0	D
1	1	1	1	1

FIG. 6

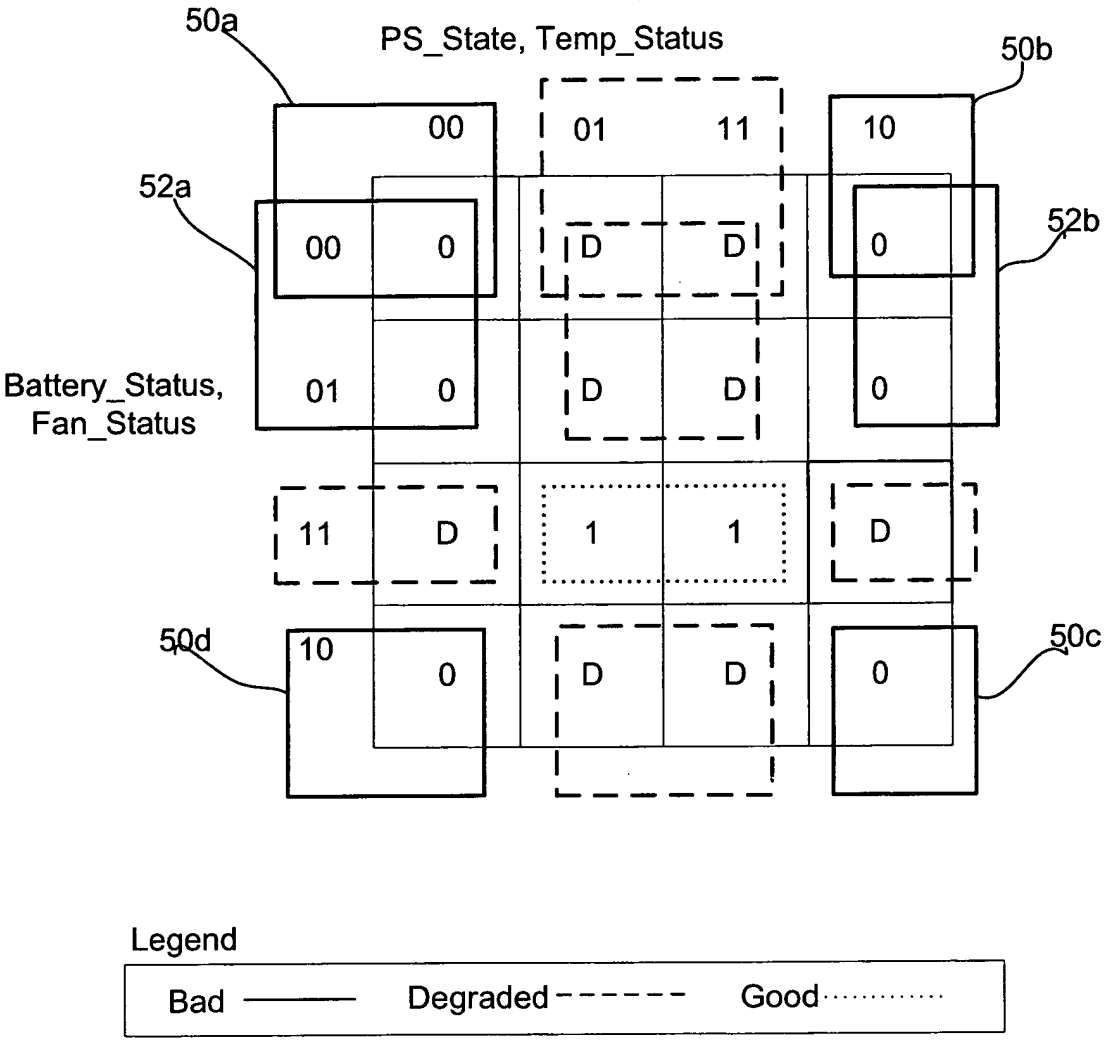

50a
PS_State, Temp_Status
50b
00
01
11
10
52a
52b
00
0
D
D
0
Battery_Status,
Fan_Status
01
0
D
D
0
11
D
1
1
D
50d
10
0
D
D
0
50c
Legend
Bad
Degraded
Good

FIG. 7

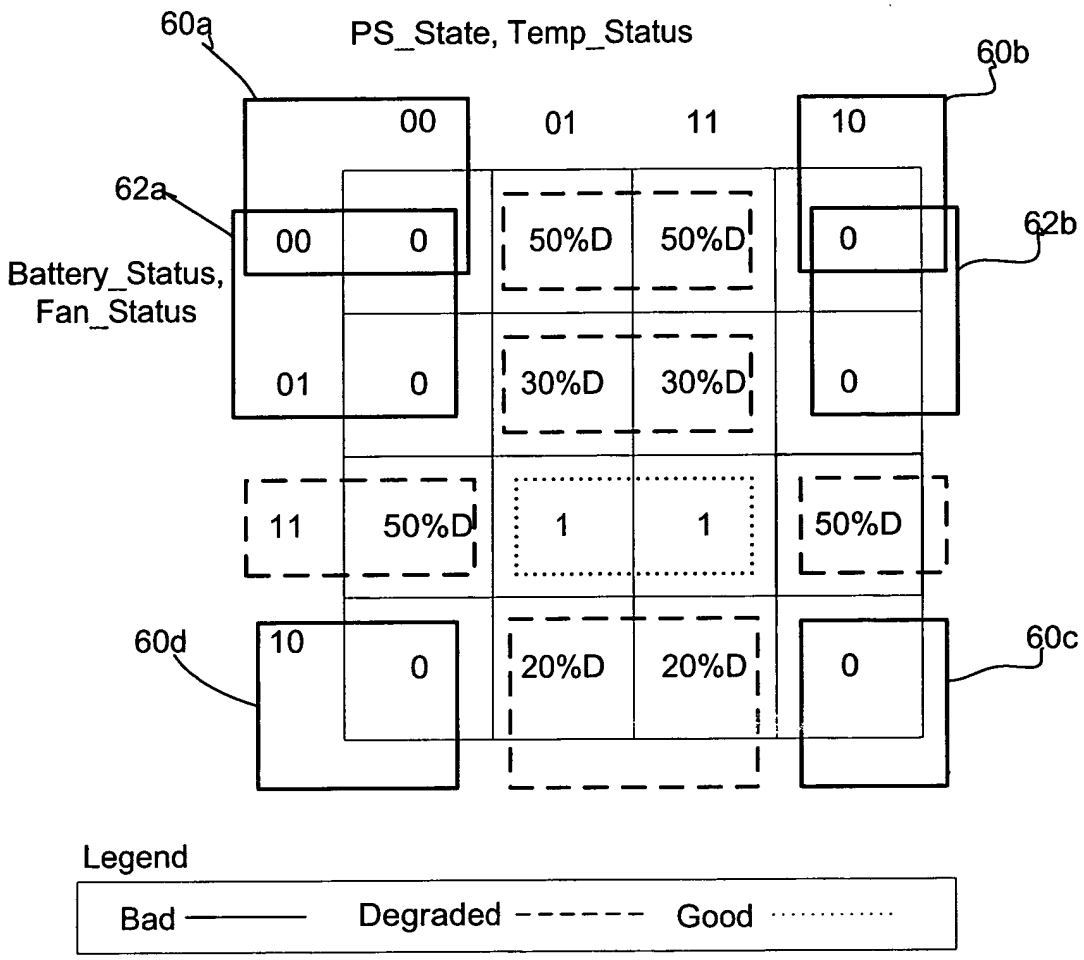
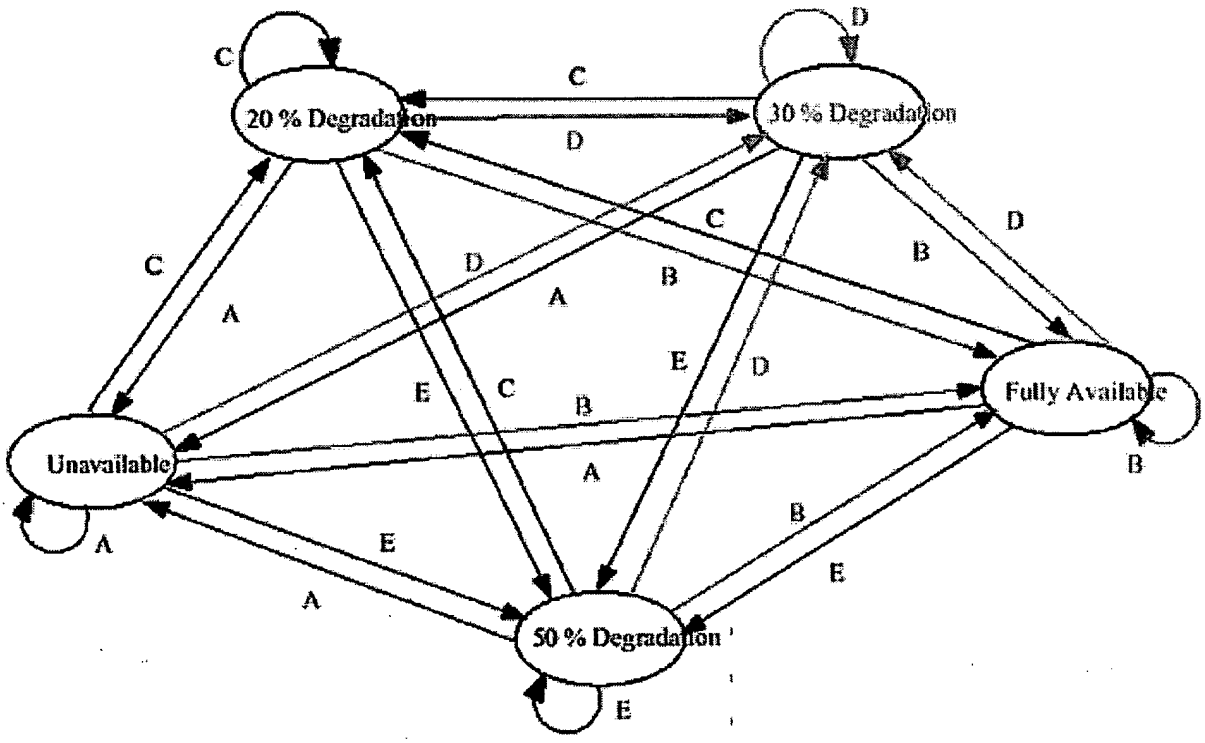

60a
PS_State, Temp_Status
60b
00
01
11
10
62a
62b
Battery_Status,
Fan_Status
00
0
50%D
50%D
0
01
0
30%D
30%D
0
11
50%D
1
1
50%D
60d
10
0
20%D
20%D
0
60c
Legend
Bad
Degraded
Good

FIG. 8



Equation A (Power Supply is Bad)	[Temp_Status] * [Battery_Status * Fan_Status]
Equation B: (Power Supply is Good)	Temp_Status * Battery_Status * Fan_Status
Equation C (20% Degradation in Power Supply)	Temp_Status * Battery_Status * [Fan_Status]
Equation D (30% Degradation in Power Supply)	Temp_Status * [Battery_Status] * Fan_Status
Equation E (50% Degradation in Power Supply)	[([Temp_Status] * Battery_Status * Fan_Status) + (Temp_Status* [Battery_Status] * [Fan_Status])

FIG. 9

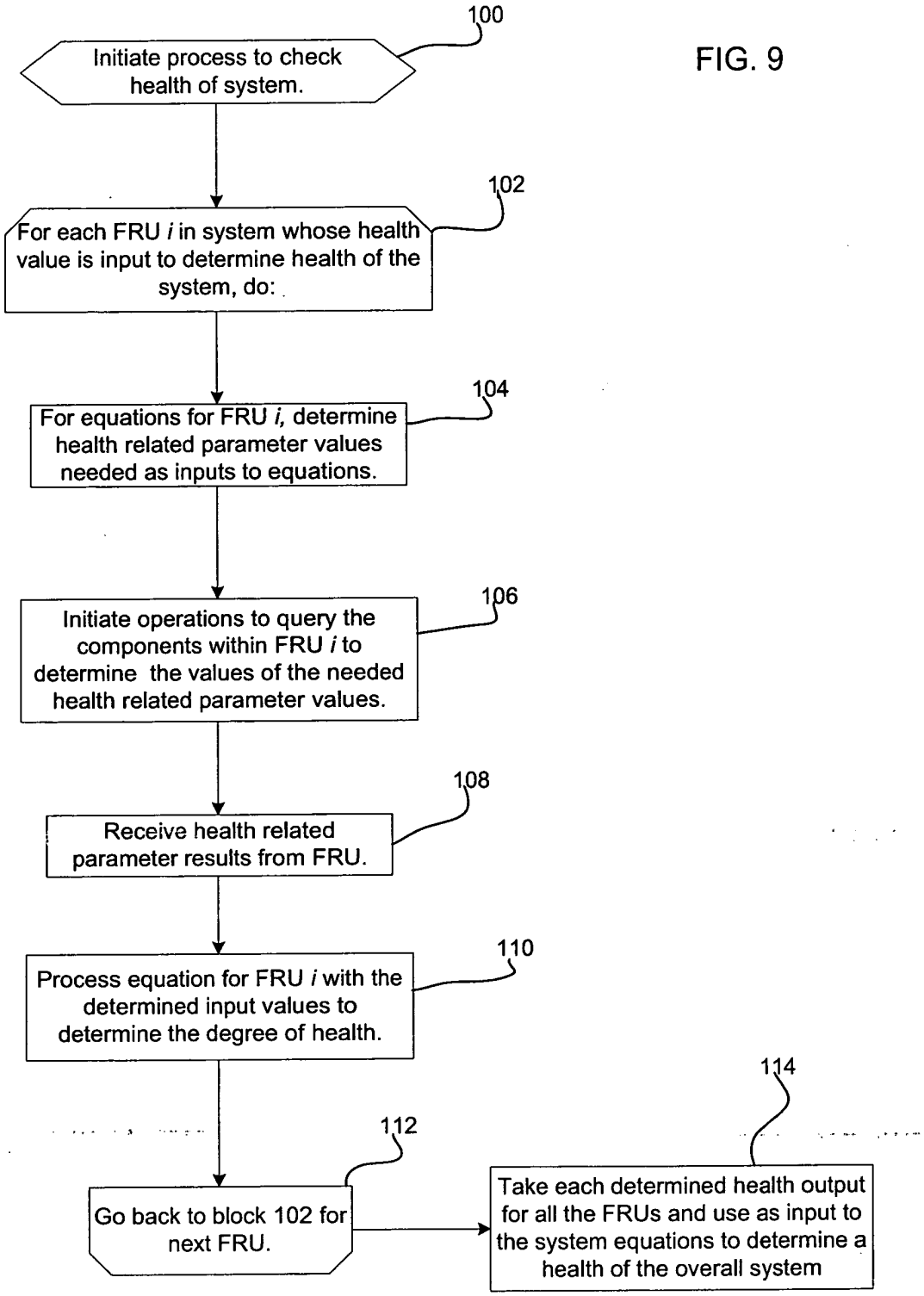

100
Initiate process to check health of system.
102
For each FRU *i* in system whose health value is input to determine health of the system, do:
104
For equations for FRU *i*, determine health related parameter values needed as inputs to equations.
106
Initiate operations to query the components within FRU *i* to determine the values of the needed health related parameter values.
108
Receive health related parameter results from FRU.
110
Process equation for FRU *i* with the determined input values to determine the degree of health.
112
Go back to block 102 for next FRU.
114
Take each determined health output for all the FRUs and use as input to the system equations to determine a health of the overall system

FIG. 10

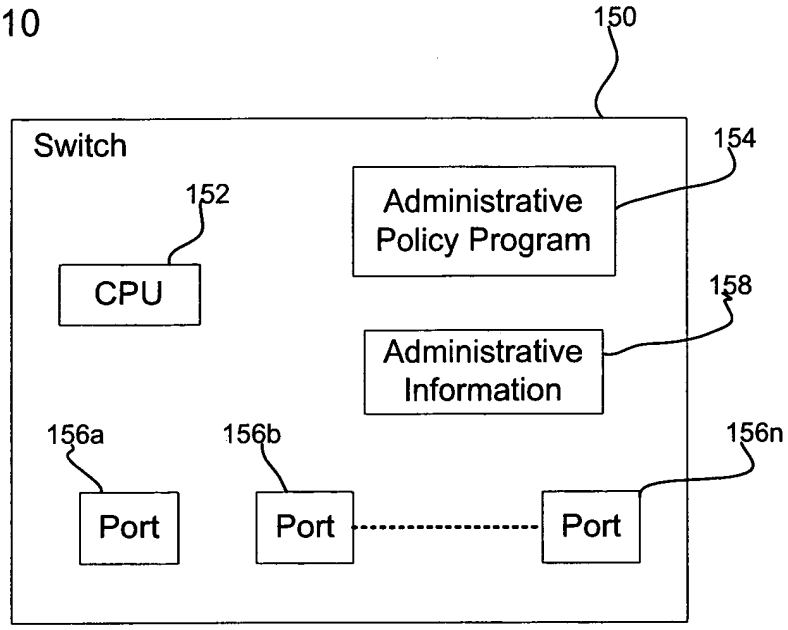

150
Switch
154
Administrative Policy Program
152
CPU
158
Administrative Information
156a
156b
156n
Port
Port
Port

FIG. 11

Administrative Policy Truth Table

Previous Admin State	*Previous Oper State*	*Current Admin State*	*Current Oper State*	*Alert Clear*	*Alert Create*	*Event*
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	1	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	0
0	1	1	0	0	1	1
0	1	1	1	0	0	1
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	1	1
1	1	1	1	0	0	0

FIG. 12

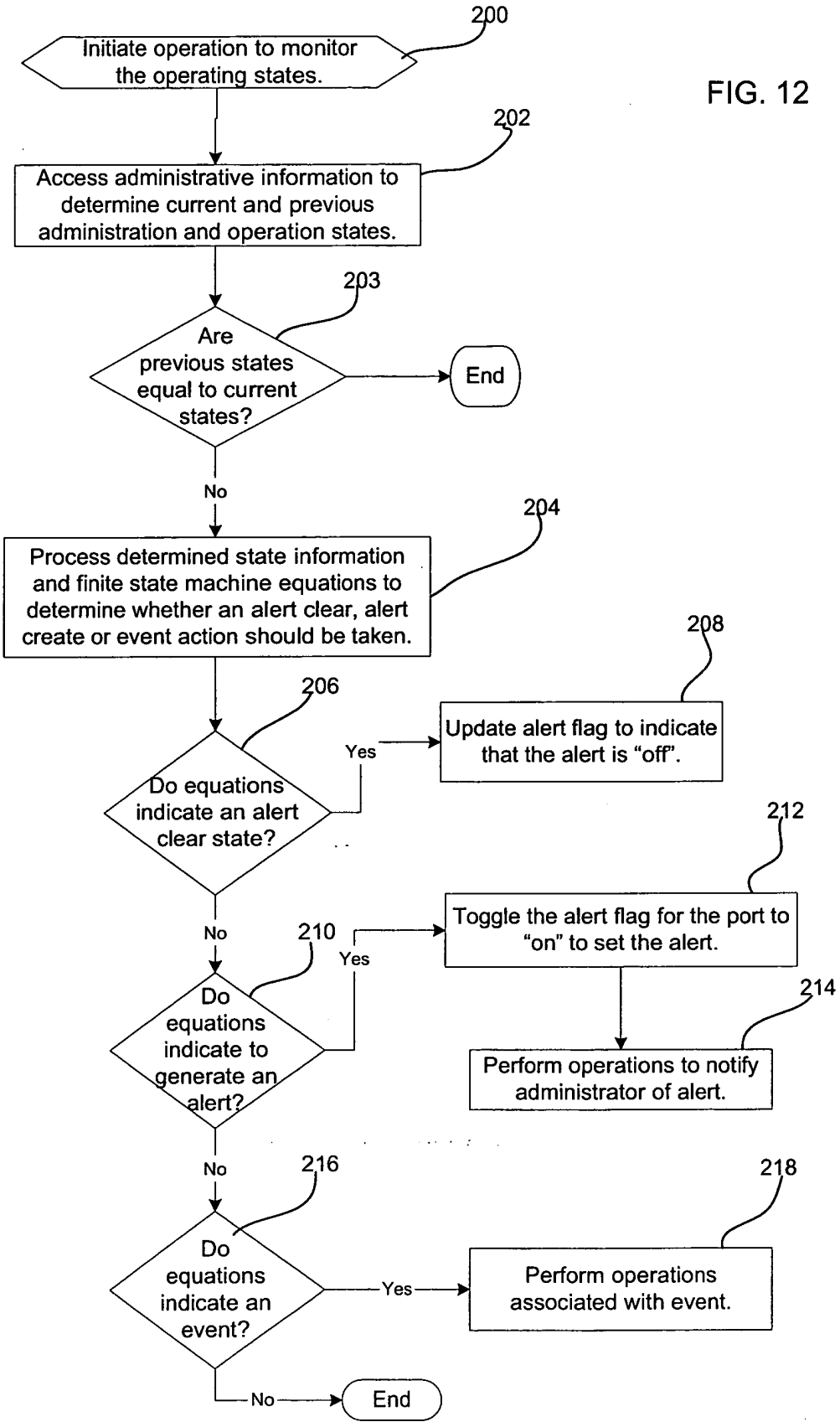
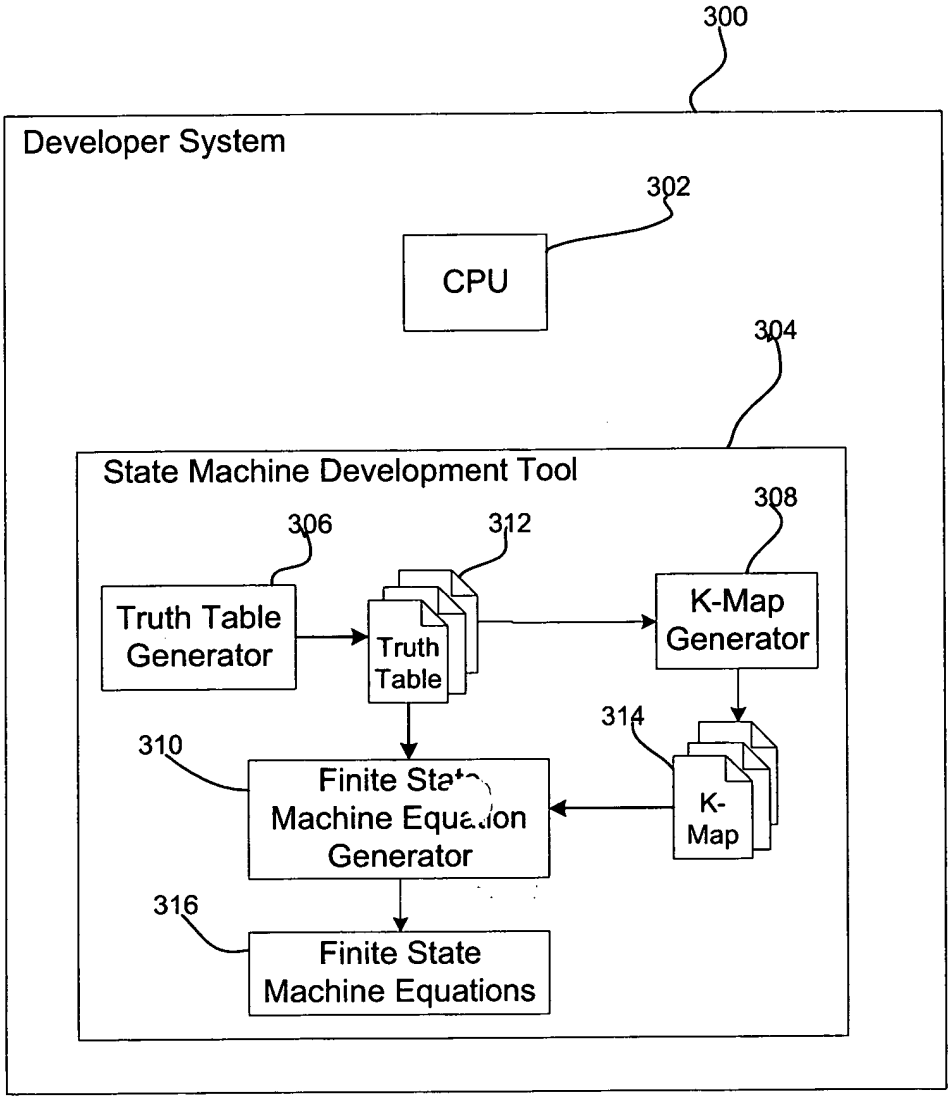

200
Initiate operation to monitor the operating states.
202
Access administrative information to determine current and previous administration and operation states.
203
Are previous states equal to current states?
End
No
204
Process determined state information and finite state machine equations to determine whether an alert clear, alert create or event action should be taken.
206
Do equations indicate an alert clear state?
Yes
208
Update alert flag to indicate that the alert is "off".
No
210
Do equations indicate to generate an alert?
Yes
212
Toggle the alert flag for the port to "on" to set the alert.
214
Perform operations to notify administrator of alert.
No
216
Do equations indicate an event?
Yes
218
Perform operations associated with event.
No
End

FIG. 13


300
Developer System
302
CPU
304
State Machine Development Tool
306
Truth Table Generator
312
Truth Table
308
K-Map Generator
314
K-Map
310
Finite State Machine Equation Generator
316
Finite State Machine Equations